Reliable Recon in Adversarial P2P Botnets

Dennis Andriesse[†], Christian Rossow[§], and Herbert Bos[†]

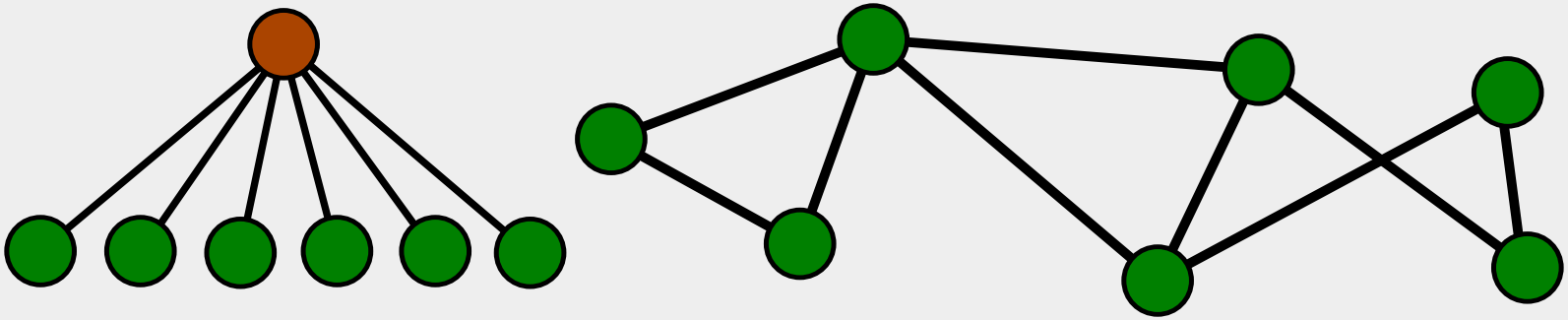
[†]Vrije Universiteit Amsterdam [§]Saarland University Germany

IMC 2015

Introduction

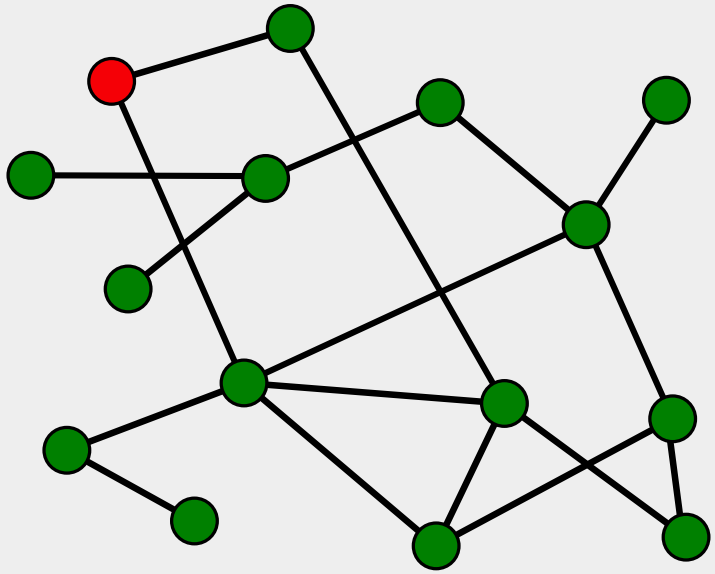
Peer-to-Peer (P2P) botnets

- Centralized botnets are vulnerable because of their C2 servers
- P2P botnets have no centralized C2 servers
  - Every bot knows some of the other bots
  - Bots use P2P communication to spread commands
  - Much more resilient against takedowns



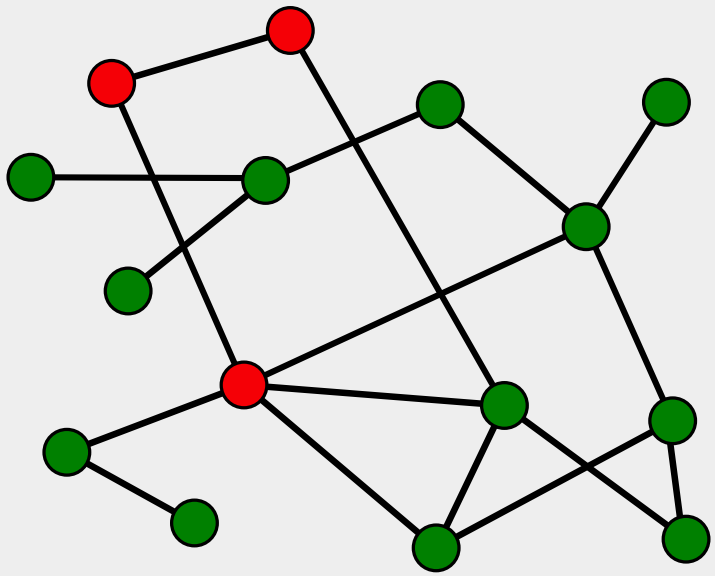
Attacking P2P botnets

- No centralized C2, must attack every bot directly
  - Report bot IPs to ISPs, poison bots, disinfect, ...
- *All attacks* (incl. recent GOZ takedown) require recon
- Most common reconnaissance strategy is *crawling*
  1. Start with a few known bots
  2. Pretend to be a neighbor and recursively ask for more bots



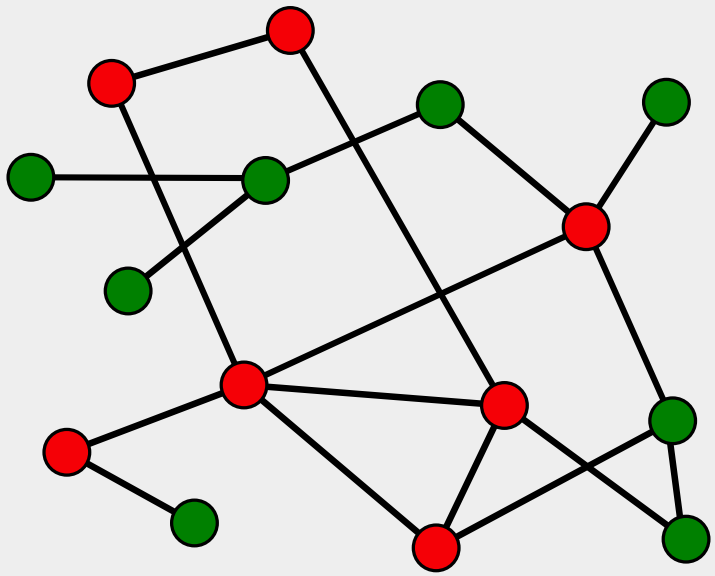
Attacking P2P botnets

- No centralized C2, must attack every bot directly
  - Report bot IPs to ISPs, poison bots, disinfect, ...
- *All attacks* (incl. recent GOZ takedown) require recon
- Most common reconnaissance strategy is *crawling*
  1. Start with a few known bots
  2. Pretend to be a neighbor and recursively ask for more bots



Attacking P2P botnets

- No centralized C2, must attack every bot directly
  - Report bot IPs to ISPs, poison bots, disinfect, ...
- *All attacks* (incl. recent GOZ takedown) require recon
- Most common reconnaissance strategy is *crawling*
  1. Start with a few known bots
  2. Pretend to be a neighbor and recursively ask for more bots



Introduction

But what if crawlers are detected?

- *Any kind of anomalous behavior can be used to detect crawlers*
- Detected crawlers are open to a multitude of attacks
  - *Blacklisting*, *retaliation*, *disinformation*, ...
  - Already observe many of these in GOZ (incl. auto-blacklisting), Sality, ZeroAccess, Hlux, ...
- We infiltrated Sality and GOZ and studied crawler quality

Detecting Crawlers

Crawler defects in GOZ and Sality

- 21 major crawlers in GOZ, 11 in Sality, *all* major protocol defects
- Operated by well-known malware analysis companies and CERTs

Defect	# of crawlers
Constrained RND/TTL/LOP/session ID	17
Low entropy session ID/bot ID/padding	10
Too many requests/only peer requests	17
Bad encryption	7

Most common defects in GOZ crawlers (more in paper)

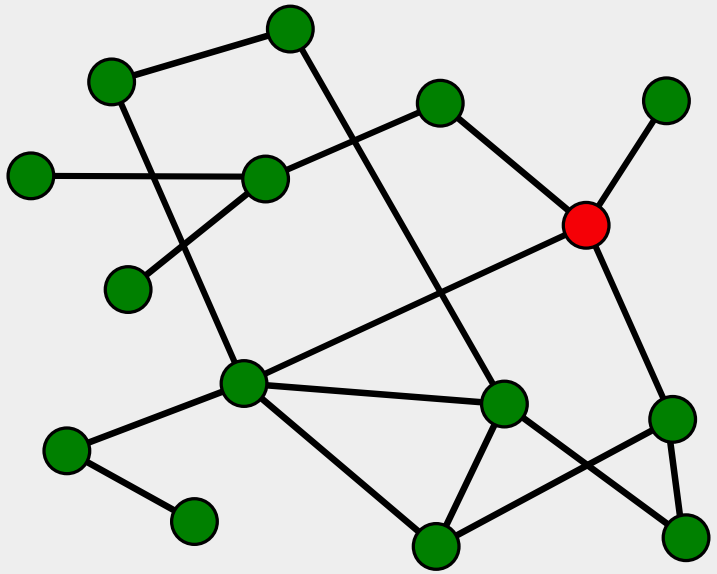
Defect	# of crawlers
Constrained LOP/port	11
Invalid version	9
Too many requests/only peer requests	11

Most common defects in Sality crawlers (more in paper)

Detecting Crawlers

Inherent crawler detectability

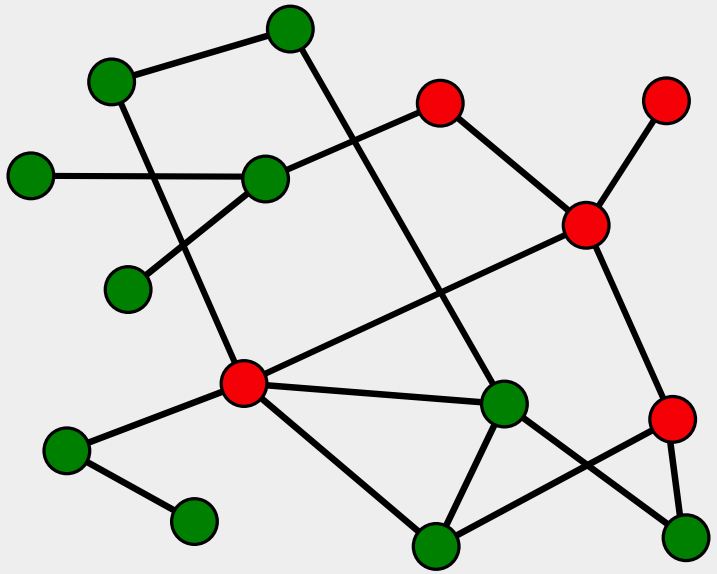
- Normal bots contact only a handful of peers (their neighbors)



Detecting Crawlers

Inherent crawler detectability

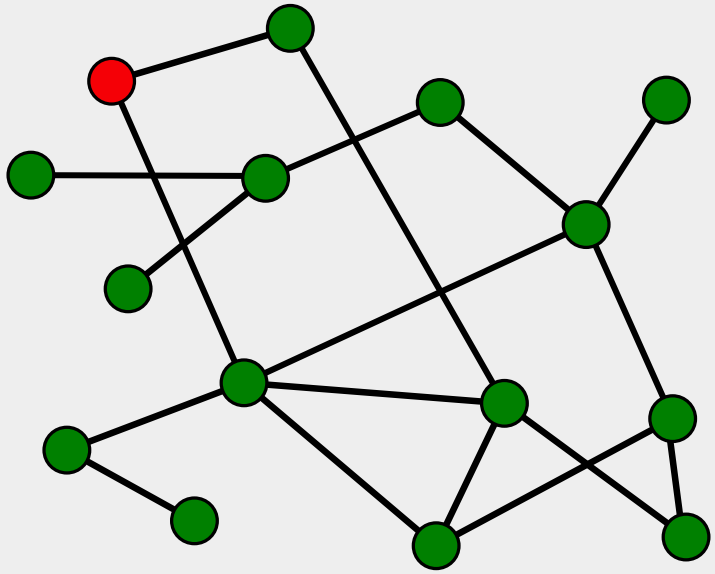
- Normal bots contact only a handful of peers (their neighbors)



Detecting Crawlers

Inherent crawler detectability

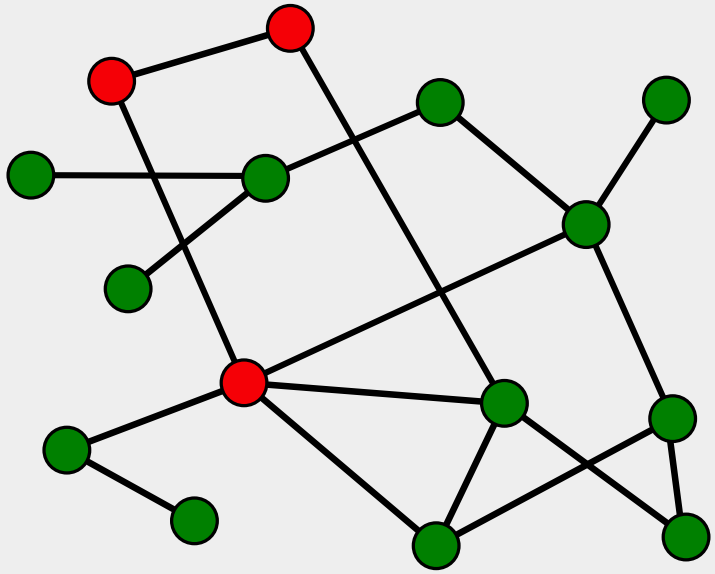
- Efficient crawlers contact (nearly) all bots to map the botnet
- This is abnormal, and *cannot be fixed* without sacrificing coverage (even minimum vertex cover may be too aggressive)
- We design an algorithm to detect crawlers by network coverage
  - Bots share who contacted them, "hard hitters" are crawlers



Detecting Crawlers

Inherent crawler detectability

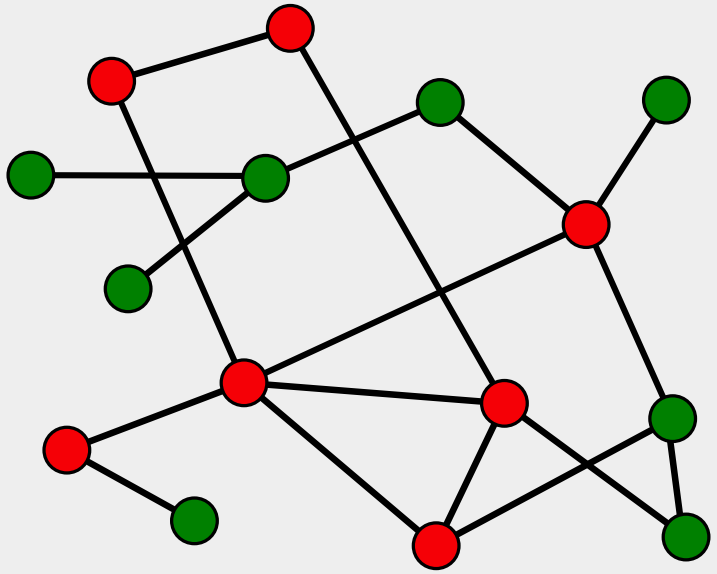
- Efficient crawlers contact (nearly) all bots to map the botnet
- This is abnormal, and *cannot be fixed* without sacrificing coverage (even minimum vertex cover may be too aggressive)
- We design an algorithm to detect crawlers by network coverage
  - Bots share who contacted them, "hard hitters" are crawlers



Detecting Crawlers

Inherent crawler detectability

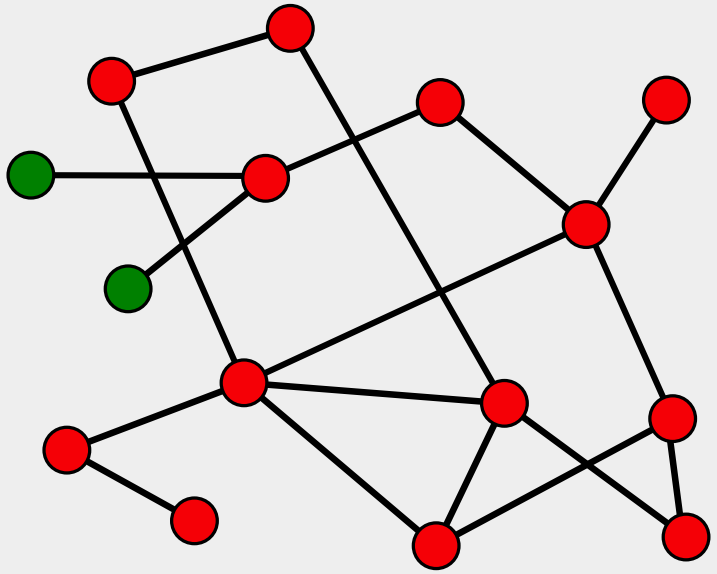
- Efficient crawlers contact (nearly) all bots to map the botnet
- This is abnormal, and *cannot be fixed* without sacrificing coverage (even minimum vertex cover may be too aggressive)
- We design an algorithm to detect crawlers by network coverage
  - Bots share who contacted them, "hard hitters" are crawlers



Detecting Crawlers

Inherent crawler detectability

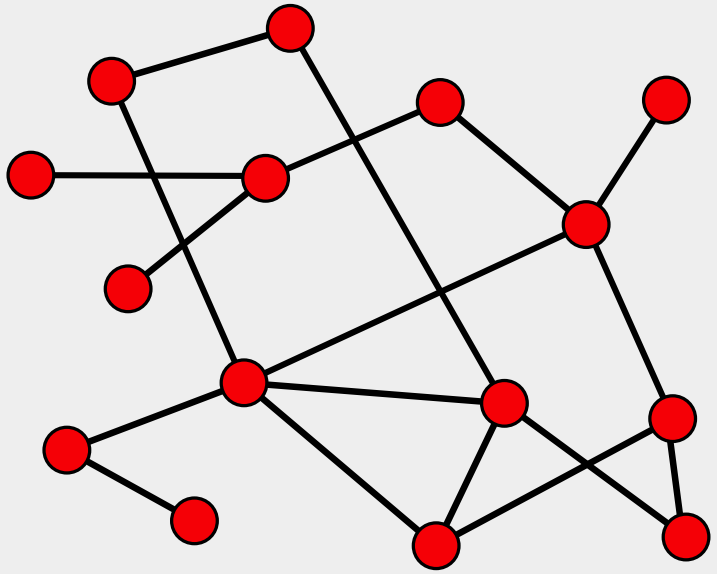
- Efficient crawlers contact (nearly) all bots to map the botnet
- This is abnormal, and *cannot be fixed* without sacrificing coverage (even minimum vertex cover may be too aggressive)
- We design an algorithm to detect crawlers by network coverage
  - Bots share who contacted them, "hard hitters" are crawlers



Detecting Crawlers

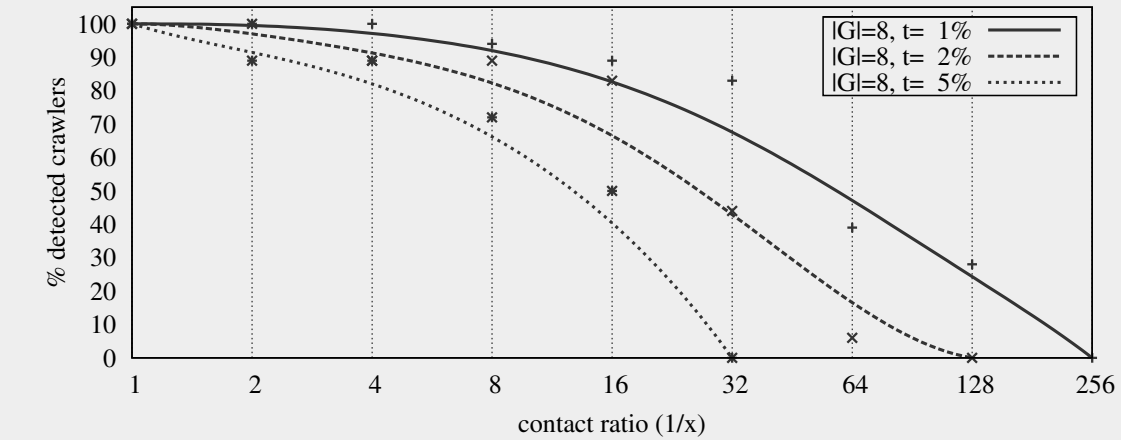
Inherent crawler detectability

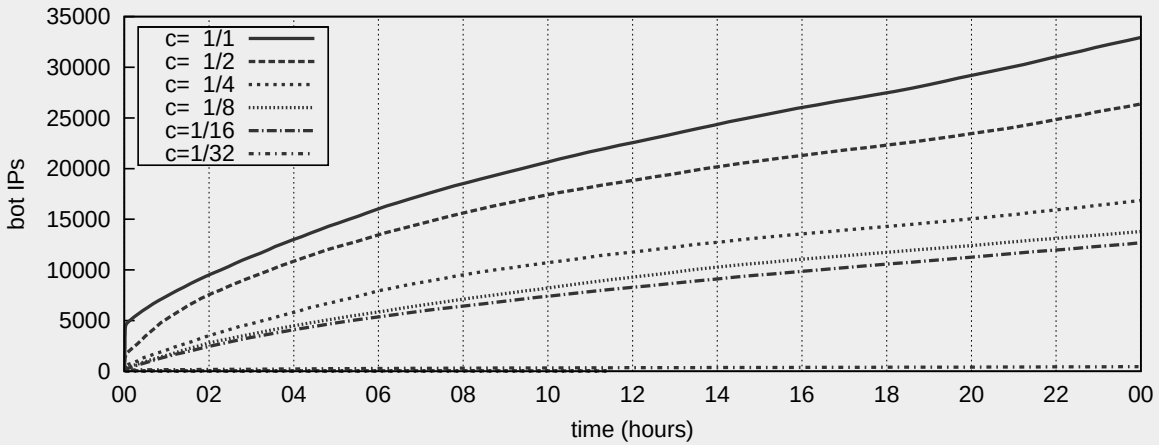
- Efficient crawlers contact (nearly) all bots to map the botnet
- This is abnormal, and *cannot be fixed* without sacrificing coverage (even minimum vertex cover may be too aggressive)
- We design an algorithm to detect crawlers by network coverage
  - Bots share who contacted them, "hard hitters" are crawlers



Avoiding detection

- Our algorithm detects all GOZ crawlers without false positives
- Crawlers must sacrifice coverage to evade detection





Alternative Recon

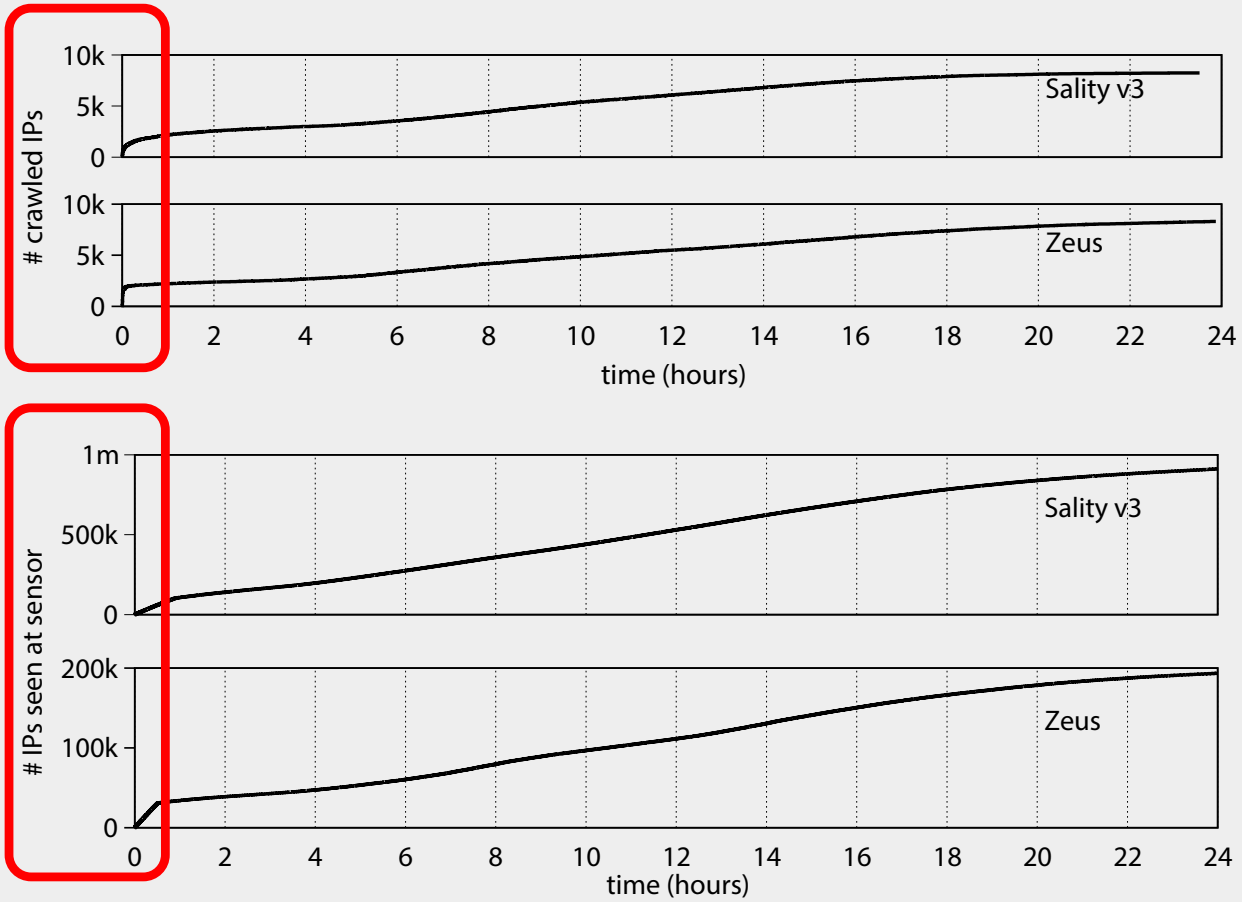
Stealthy Crawling

- Contact Ratio Limiting/Request Frequency Limiting
  - Performance/coverage issues (see previous slide)
- Distributed Crawling (distribute/rotate egress traffic source IPs)
  - Works for GOZ given ≥ 32 distinct /20's, or a /16
- Anonymizing Proxies (with fast IP rotation)
  - Feasible given sufficient network block (which may not leak)

Alternative Recon

Passive Sensors

- Far better coverage than crawlers (no NAT/firewall issues)
- In contrast to crawlers, sensors verify authenticity of each bot



Alternative Recon

Internet-Wide Scanning

- Proposed as alternative recon strategy, e.g. for ZeroAccess
- Does **not** generalize
  - Port range often too large to scan
  - Suitable probes may not exist (e.g., due to encryption etc.)
  - NAT traversal issues

	Fixed port	**Probe msg**	**Susceptible**
GOZ	✗	✗	✗
Sality	✗	✓	✗
ZeroAccess	✓	✓	✓
Kelihos/Hlux	✓	✓	✓
Waledac	✗	✓	✗
Storm	✗	✓	✗

Susceptibility of P2P botnets to Internet-wide scanning

Conclusion

Where to go from here?

- Crawlers are most popular recon, but offer poor stealth/coverage
- *All efforts against P2P botnets hinge on reliable recon*
- **Fix your crawlers, or switch to alternatives!**